Daten bändigen & visualisieren mit R

Was wir machen und wie wir uns organisieren

B. Philipp Kleer
Methodentage 2021
11. Oktober 2021

Was ist tidyverse?

Tidyverse ist ein Paket, dass mehrere Pakete beinhaltet, die alle nach ähnlicher Syntax funktionieren und untereinander kompatibel sind.

Es bietet somit einen sehr großen Funktionsumfang und wird daher auch viel genutzt.

`

The Core tidyverse

Tidyverse beinhaltet Kernpakete, die allesamt mit dem Befehl `library("tidyverse")` geladen werden. Dies sind:

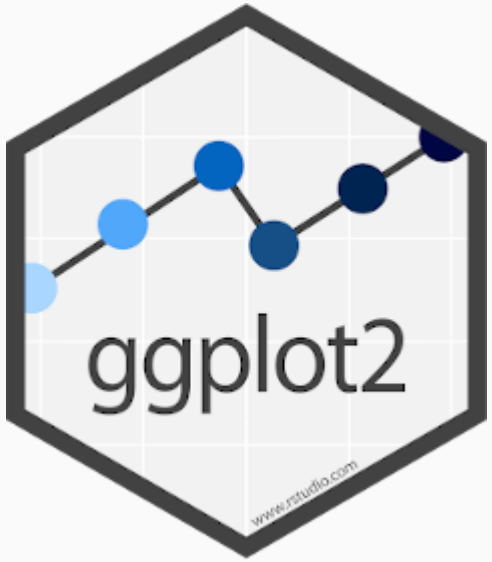
- **dplyr** (Datenbereinigung)
- **ggplot2** (Grafiken)
- **stringr** (Umgang mit Textdaten)
- **tidyr** (Umgang mit Datensätzen)
- forcats (Umgang mit Faktoren)
- tibble (Tabellentool)
- readr (Import von Daten)
- purrr (Umgang mit Funktionen und Vektoren)

`

Vier Pakete im Fokus

Cheat-Sheet

Cheat-Sheet



Cheat-Sheet

Cheat-Sheet

`

Ziel des Workshops

Die Teilnehmenden können am Ende des Workshops ...

- ... die Grammatik der Pakete ggplot2 und tidyverse verstehen und auf eigene Zwecke anwenden.
- ... Daten zielführend aufbereiten.
- ... Daten und Ergebnisse sinnvoll darstellen.
- ... erste eigene Funktionen programmieren.

`

Ziel des Workshops

Es geht in diesem kurzen 1-Tages-Kurs vor allem um **Readbility-Skills**. Ziel ist es, dass man neue Probleme mit dem hier gezeigten lösen kann. Dafür sollte der Inhalt aber während des Workshops gut aufbereitet bzw. nachbereitet werden (eigene Notizen in den Skripten etc.).

In meinen Kursen gilt immer folgendes Prinzip: **Was man nicht schreibt, lernt man auch nicht!**

Es ist also eine didaktische Entscheidung von mir, dass es keine fertige Skripts gibt. Selbst programmieren heißt eben auch selbst Code schreiben und nicht nur einzelnen Felder austauschen (wie bei *Click and Play* mit SPSS). Ebenso wird meiner Meinung nach das Verständnis von Funktionen viel besser vermittelt.

Die Slides bzw. HTML-Dokumente haben aber kopierbaren Code (ein bisschen Erleichterung). Wenn man in diesen Dokumenten über den Code geht, erscheint oben rechts ein *Clipboard*, mit dem man den Code in die Zwischenablage kopiert.

```
install.packages("tidyverse")
library("tidyverse")

# alternativ:
# install.packages("dplyr")
# library("dplyr")
```

`

Wer ich bin und wie ich Workshops leite?

Wer bin ich?

- seit 2015 Mitarbeiter an der Professur für Methoden (viele praktische Methoden-/Projektkurse bisher gegeben)
- nutze seit mehreren Jahren bereits R bzw. RStudio (vor allem Projekt-Funktion
- derzeit: gefördertes Lehrprojekt, in dem R-Kursmaterial für Personen aufbereitet wird, die keine Computer-/Programmierkenntnisse haben

Wie ich meine Rolle als Workshopleiter sehe?

1. kollegiales, respektvolles Mitaneinander
2. Interesse daran, anderen zu helfen/zu unterstützen
3. in der Ansprache ziehe ich das Du vor
4. Kurz-Inputs und dann eigenes *trial-and-error*
5. Unterstützung beim Code-Crashing
6. gebe (viel) Input, aktive Mitarbeit aber erforderlich

Alle Kursmaterialien sind entweder auf **gitlab** oder in der **R Studio Cloud** runterzuladen. In **ILIAS** finden sich Links an die betreffenden Stellen.

`

Arbeiten mit R

Arbeiten mit R heißt in der Regel immer wieder auf Probleme zu stoßen und willens zu sein, diese Probleme zu lösen. In meiner jetzt fast zehnjährigen Arbeit mit R bin ich noch nie auf ein Problem gestoßen, dass man nicht lösen konnte (auch wenn es manchmal umständlich war).

Wichtig dafür: Lesefähigkeit von Code. Also das Verständnis von Code. Dies ist auch das primäre Ziel des heutigen Tages.

Wichtig für Lesefähigkeit: Sauberer Code! Denn: *In every project you have at least one other collaborator: future-you. You don't want future-you to curse past-you!*

Ebenso hilfreich: Projekte in RStudio nutzen (das machen wir in RStudio Cloud) oder sogar mit **R Notebooks** arbeiten

`

Kursmaterialien

Wie gesagt, sind die Kursmaterialien direkt in der **RStudio Cloud** verlinkt, aber auch in **gitlab**.

Die `code chunks` haben ein integriertes *Clipboard*, mit dem der Code direkt in ein R Skript kopiert werden kann. Dafür geht man einfach beim betreffenden Code oben rechts auf das *Clipboard*-Zeichen.

```
install.packages("tidyverse",
                 dependencies = TRUE)
```

Wichtig: Es handelt sich um *.html*-Präsentationen, die grafisch nur dann korrekt angezeigt werden, wenn eben auch die anderen Ordner relativ genauso lokal gespeichert sind, wie es beim Download entsteht.

`

Coding Konvention

Jede Person hat eigene Vorlieben, was den geschriebenen Code angeht. Im Folgenden möchte ich nur kurz meine Präferenzen darlegen.

Für neue Variablen oder Dataframes nutze ich in der Regel das Format `lowerCamelCase`:

```
df$newVar <- NA

newDf <- subset(df,
                is.na(newVar
                      )
                )
```

Selbst geschrieben Funktionen schreibe ich mit `_`:

```
own_mean <- function(x){
  mean = sum(x) / length(x)
  print(mean)
  }
```

`

Code Konvention

Wie im ersten Fall bereits sichtbar, trenne ich Argumente mit `Enter` (Zeilenumbruch) und setze Klammern ebenfalls in neue Zeilen. Das hat den Vorteil, dass die Kommentierung einfacher erfolgen kann. Hat eine Funktion nur ein einziges Argument bleibt die Klammer in der gleichen Zeile.

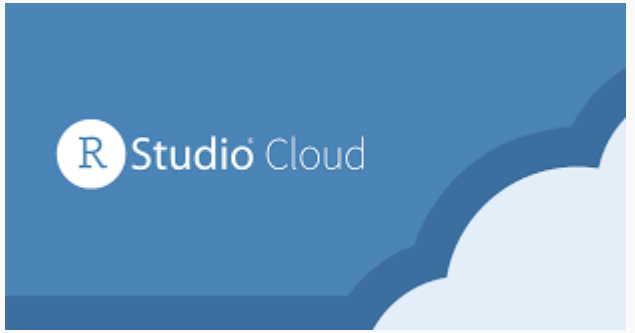
```
# einzelnes Argument
library(tidyverse)

# mehrere Argumente
str_sub(tweet$text[23], # Quelle
        -20,  # Beginn 20. Buchstaben vom Ende
        -2    # vorletzter Buchstabe vom Ende
        )
```

`

Handling in RStudio Cloud

Link zur RStudio Cloud

Falls noch nicht registriert, bitte registrieren und dann Space beitreten.



- browserbasierte Version von RStudio Cloud
- kein leistungsstarker Rechner nötig
- Teilen von Code ist erleichtert (*sharing projects*)
- Projektbasierte Ordner

`

Start

Der Kurs setzt Grundkenntnisse voraus. Ihr lernt euch jetzt in Breakout-Rooms kennen. In den Breakout-Rooms sollt ihr euch kennenlernen und ein paar Grundaufgaben in R lösen. Dies dient auch der Auffrischung. Ich schaue abwechselnd in den Breakout-Rooms nach. Ihr könnte mich aber auch rufen.

Folgende Aufgaben sind zu erledigen:

1. Den Datensatz `pss.rds` in das environment laden (fiktiver Datensatz Panem Social Survey)
2. Die Variable `agea` deskriptiv beschreiben.
3. Schafft eine neue Variable, die dem Datensatz hinzugefügt werden soll. Diese Variable soll `socgroup` heißen und einfach eine Sequenz von `1, 2, 3, 4` über die Länge des Datensatzes beinhalten. **Wichtig**: Jede Zahl soll gleich oft vorkommen. Ob sich die Sequenz immer wiederholt (also Reihenfolge `1, 2, 3, 4, 1, 2, 3, 4, ...`), oder ihr erst alle `1`, dann alle `2` etc. abbildet, ist euch überlassen.
4. Vergesst nicht, euch gegenseitig vorzustellen!

Zeit: 30 Minuten.

Wenn jemand nicht seine lokale R-Installation nutzen möchte, kann er einfach auf die **RStudio Cloud** zurückgreifen. Dort sind die Datensätze, Skripte & Folien auch bereits hinterlegt und müssen nicht direkt runtergeladen werden.

Das war's!